AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method comprising:

building a queue having one or more drivers for execution prior to booting an operating system; and

executing the one or more drivers in the queue using a plurality of processors, the plurality of processors including a bootstrap processor and one or more application processors, wherein the execution of drivers by each of the plurality of processors includes:

determining whether there is a first driver of the one or more drivers in the queue,

determining whether the first driver is ready for execution, and

if the first driver is ready for execution, removing the first driver from the queue and executing the first driver.

2-3. (Cancelled)

4. (Currently amended) The method of ~~claim 2~~ claim 1, wherein if ~~a second~~ a first application processor ~~in the plurality~~ of the one or more application processors determines that there are no drivers left in the queue, the ~~second~~ first application processor goes to an idle state.

5. (Currently amended) The method of claim 4, wherein if the ~~first~~ bootstrap processor determines there are no drivers left in the queue, the ~~first~~ bootstrap processor:

waits until all other processors in the plurality of processors are in an idle state; and

boots [[an]] the operating system.

6. (Original) The method of claim 1, wherein the plurality of processors includes one or more logical processors.

7. (Cancelled)

8. (Currently amended) The method of claim 1, wherein the ~~method is utilized in an extensible firmware interface (EFI)~~ each of the plurality of drivers is compatible with an Extensible Firmware Interface (EFI) specification.

9. (Original) The method of claim 1, wherein the plurality of drivers are executed in order.

10. (Original) The method of claim 9, wherein a first driver in the queue that has a dependency on a second driver in the queue is not executed until the second driver has been executed.

11. (Currently amended) A processor comprising:

an execution unit; and

a first logical processor and a second logical processor, the first logical processor and the second logical processor utilizing the execution unit;

the first logical processor to build a queue having one or more drivers to be executed prior to booting an operating system; and

the first logical processor and the second logical processor to execute the one or more drivers in the queue in parallel at least in part, wherein the execution of drivers includes:

determining whether there is a first driver of the one or more drivers in the queue,

determining whether the first driver is ready for execution, and

if the first driver is ready for execution, removing the driver from the queue and executing the first driver.

12. (Original) The processor of claim 11, wherein if the second logical processor determines there are no drivers left in the queue, the second logical processor is to enter an idle state.

13. (Currently amended) The processor of claim 12, wherein if the first logical processor determines there are no drivers left in the queue, the first logical processor is to:

wait until the second processor is in an idle state; and

boot [[an]] the operating system.

14. (Original) The processor of claim 11, wherein the processor operates concurrently with one or more other processors.

15. (Cancelled)

16. (Currently amended) A system comprising:

a bootstrap processor;

one or more application processors;

a bus, the bootstrap processor and the one or more application processors being coupled to the bus; and

a flash memory coupled to the bus;

wherein prior to the booting of an operating system for the system the bootstrap processor and the one or more application processors execute a plurality of drivers in a queue in parallel at least in part, the execution of the drivers by the bootstrap processor and each of the one or more application processors including:

determining whether there is a first driver of the plurality of drivers in the queue to be executed,

determining whether the first driver is ready for execution, and

if the first driver is ready for execution, removing the first driver from the queue and executing the first driver.

17. (Currently amended) The system of claim 16, wherein if [[an]] a first application processor of the one or more application processors determines that there are no drivers left in the queue, the first application processor is to enter an idle state.

18. (Currently amended) The system of claim 17, if the first processor determines there are no drivers left in the queue, the first processor is to:

wait until all of the one or more application processors are in an idle state; and

boot [an] the operating system.

19. (Cancelled)

20. (Currently amended) The system of ~~claim 19~~ claim 16, ~~the method is utilized in an extensible firmware interface (EFI)~~ wherein each of the plurality of drivers is compatible with an Extensible Firmware Interface (EFI) specification.

21. (Currently amended) A ~~machine-readable~~ computer-readable medium having stored thereon data representing sequences of instructions that, when executed by a processor, cause the processor to perform operations comprising:

building a queue having one or more drivers to be executing prior to booting of an operating system; and

executing the one or more drivers in the queue using a plurality of processors, the plurality of processors including a bootstrap processor and one or more application processors, wherein the execution of drivers by each of the plurality of processors includes:

determining whether there is a first driver of the plurality of drivers in the queue,

determining whether the first driver is ready for execution, and

if the first driver is ready for execution, removing the first driver from the queue and executing the first driver.

22-23. (Cancelled)

24. (Currently amended) The medium of ~~claim 22~~, claim 21 wherein if a ~~second~~ first application processor of the ~~plurality of~~ one or more application processors

determines that there are no drivers left in the queue, the ~~second~~ first application processor goes to an idle state.

25. (Currently amended) The medium of claim 24, wherein if the ~~first~~ bootstrap processor determines there are no drivers left in the queue, the ~~first~~ bootstrap processor is to:

wait until all other processors in the plurality of processors are in an idle state;

and

boot [[an]] the operating system.

26. (Original) The medium of claim 21, wherein the plurality of processors includes one or more logical processors.

27. (Cancelled)

28. (Currently amended) The medium of claim 21, wherein the ~~method is utilized in an extensible firmware interface (EFI)~~ wherein each of the plurality of drivers is compatible with an Extensible Firmware Interface (EFI) specification.

29. (Original) The medium of claim 21, wherein the plurality of drivers are executed in order.

30. (Original) The medium of claim 29, wherein a first driver in the queue that has a dependency on a second driver in the queue is not executed until the second driver has been executed.